Appendix 2: The *DMRScan* R package

## Introduction

The package is placed on the R repos Bioconductor for the stable release, and on github for a development version. To install the package:

source(<https://bioconductor.org/biocLite.R>)  
biocLite("DMRScan") #For the stable release or

The main function is the sliding window DMRScan(). The input to this function is a set of potential clusters (eg collections of CpGs in close proximity or in a set of genes or features). This object is obtained using the makeCpGregions() or makeCpGgenes() with a list of chromosomal positions, the maximum distance between two CpGs within a cluster, and the minimum number of CpGs in a cluster or gene as its input.

## Pipeline

### 01 Load the data and calculate a test statistic for each CpG

library(DMRScan)   
## Load methylation data from chromosome 22, with 52018 CpGs measured  
data(DMRScan.methylationData)  
data(DMRScan.phenotypes) ## Load phenotype (end-point for methylation data)  
## Do logistic regression on each CpG to get a test statistic:   
observations <- apply(DMRScan.methylationData,1,function(x,y){summary(glm(y ~ x, family = binomial(link = "logit")))$coefficients[2,3]}, y = DMRScan.phenotypes)  
head(observations)

### 02 Generate CpG clusters

## pos is the physical chromosomal location for each probe or CpG  
pos <- matrix(as.integer(unlist(strsplit(names(observations),split="chr|[.]"))), ncol = 3, byrow = TRUE)[,-1]  
regions <- makeCpGregions(observations = observations, chr = pos[,1], pos = pos[,2], maxGap = 750, minCpG = 3)

### 03 Estimate window thresholds for different window sizes

When only one window size is specified, then this is equivalent to the *R* test from Zhang (2004), and when a sequence of window sizes is specified then the *S* test from Zhang (2004) is used. The window sizes reflect the number of probes or CpGs within a window, independent on the distance between them. This is handled by the makeCpGregions() which sets the maximum allowed distance between the CpGs in one cluster or region.

## The window size can be either a single window size, or a sequence of window sizes.   
window.sizes <- 3:7 ## Number of CpGs in the sliding windows

n.CpG <- nCpG(regions) ## Total number of CpGs to be tested

Now we can estimate the window threshold using three different methods, described in detail in the paper: MCMC with (almost) any correlation structure between each CpG, Importance Sampling with a more restricted dependency structure or a closed form approximation to the thresholds assuming that the test statistic for the CpGs follows an OU-process.

## Estimate the window threshold, based on the number of CpGs and window sizes using important sampling:  
window.thresholds.importancSampling <- estimateWindowThreshold(nProbe = n.CpG, windowSize = window.sizes, method = "sampling", mcmc = 10000)

## And estimate the window threshold using the closed form expression from Siegmund  
window.thresholds.siegmund <- estimateWindowThreshold(nProbe = n.CpG, windowSize = window.sizes, method = "siegmund")

## Estimating using MCMC can take quite a while to do, and do not perform much better than the Importance sampling. The code model parameter is the model that is passed on to arima() for simulated a time series of observations.

window.threshold.mcmc <- estimateWindowThreshold(nProbe = n.CpG, windowSize = window.sizes, method = "mcmc", mcmc = 1000, nCPU = 1, submethod = "arima", model = list(ar = c(0.1,0.03), ma = c(0.04), order = c(2,0,1))) ## This can take quite a long time to run.

### 04 Identifying differentially methylated regions using DMRScan()

The input to DMRScan is the region list from makeCpGregions(), the window sizes and the estimated window thresholds. To estimate the DMRs using the Importance sampling threshold calculated in the previous section:

dmrscan.results <- DMRScan(observations = regions, windowSize = window.sizes, windowThreshold = window.thresholds.importancSampling)  
print(dmrscan.results) )## Print the result